DARPA Success Story

How Agilent VEE was Utilized in the Development of a Motorcycle Designed for a Tough 250-Mile Race – No Driver Allowed

Teaming a driverless motorcycle with the wits of a crackerjack graduate and undergraduate college design group, enabled the development of a fly-by-wire, fully-autonomous vehicle able to conquer the rugged terrain of the Mohave Desert from Barstow to Las Vegas

By Anthony Levandowski,
University of California at Berkeley

Designing a vehicle to navigate a 250-mile path, over difficult terrain in the Mohave Desert, controlled by navigation that relies on GPS and on-board sensors to alert the vehicle to obstacles along the way, is no trivial task. And if you choose as your design team a diverse group speaking eight languages and five engineering and scientific disciplines and a shoestring budget – that too adds to the challenge. Then if you pick as your vehicle a two-wheeled motorcycle, as depicted in **Figure 1**, rather than a more conventional off-road vehicle, such as a Hummer, that adds still more challenges.


Figure 1.

Narrow Aperture – By choosing a motorcycle, rather than a 4-wheel vehicle, the Cal-Berkeley team had greater tolerances for staying within boundaries required by the contest and navigating narrow paths such as tunnels

The ‘Grand Challenge’ was a race sponsored by the Defense Advanced Research Projects Agency (DARPA), an agency of the Department of Defense. It pitted vehicles against each other that were developed by some of the best engineering talent at leading universities. The path was from Barstow, California to Las Vegas, NV, approximately 250 miles long, with a one-million dollar cash prize awarded to the first team that completed the race in an off-road vehicle in under 10 hours. The Grand Challenge took place on March 13th, 2004.

Our development team comprised twenty-one Cal-Berkeley graduate and undergraduate students, with assistance from several facility advisors. The group included people from all over the world, speaking eight different languages, each with a different background. We were staffed with team members who specialized in biochemistry, applied math, programming – as well as electrical and mechanical engineering.

What's the Grand Challenge?

Here is why this race posed such a challenge:

- The course the vehicles were to follow was not disclosed to the contestants until two hours before the start of the race. It meant that each vehicle had to be ready and able to successfully take on all types of terrain.
- The vehicle had to be fully autonomous. It could not be remote controlled. So it demanded that the contestants devise highly-innovative sensor technology and new control platforms to keep their vehicle on course, yet avoid ground obstacles along the way.
- There was an opportunity to refuel half way, but the rules dictated that it had to be a totally autonomous servicing and refueling. We decided to avoid this by not refueling at all. We realized 50 mpg (miles per gallon) on our motorcycle – versus 6 mpg for a Hummer. Our motorcycle came equipped with a 2.4-gallon tank which took us half way. By strapping on another 2.5-gallon tank, we could traverse the 250-mile course without refueling.

The path was described in a route definition data file. This file comprised a set of latitude and longitude coordinates as well as a radius about each which were the leg boundaries. Each vehicle was to remain, at all times, within the corridors defined by the inner-most boundary. When a small way point was followed by a large way point, the vehicle was required to stay within the boundaries of the smaller path while intersecting and connecting with the next way point.

The kinds of terrain our vehicle was expected to encounter included:

- A water crossing
- An underpass ten feet wide by nine feet high
- An overpass
- Trail obstacles, such as puddles, rocks, etc.

Why We Chose a Motorcycle

Once we began to think about what the best platform would be for an off-road vehicle, it became clear that the answer was a motorcycle. Our entry was the world's first drive-by-wire autonomous motorcycle. Here are some of the reasons behind our choice:

- **Cheaper** – we simply couldn't afford a Hummer to test our systems.
- **Smaller** – you could drop it off an airplane, it was a more compact, narrow-aperture package.
- **More capable** – running at 60 mph would be very risky if you were in a four-wheel vehicle since there would be a high likelihood of flipping over. Whereas a two-wheel platform could safely traverse the periphery of a cliff, from one end to the other.
- **Lighter** – therefore the forces involved, should our vehicle crash, were much smaller. The motorcycle weighed 220 lbs, so if it crashed, it was less likely to be damaged.

One of the major challenges was to stabilize our motorcycle while it was standing still as well as when under way. To keep the vehicle upright while stationary, we needed to be able either to shift a mass traversely across the vehicle or apply a moment. We decided to employ a control moment gyroscope (CMG), shown in **Figure 2**, that would apply a moment traversely across the vehicle. If the motorcycle began to fall to the left, we could develop and apply a counteracting force to the right.

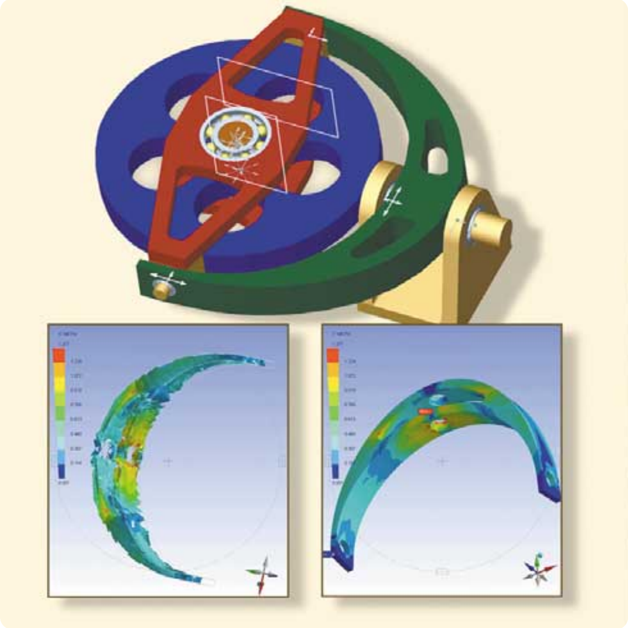


Figure 2.
A control moment gyroscope provided the necessary moment to right the motorcycle when on the ground and to stabilize it while underway

Thanks to the CMG the bike would stand upright at the starting line. If it flew into the air once it was under way, it would remain at the same angle it was at when it left the ground. With the CMG, we could also right our motorcycle on any kind of surface. It didn't have to be flat.

We knew our vehicle probably would encounter shrubbery, bushes, water and slippery terrain. It might also need to traverse a 45-degree slope. In so doing, it might slide, but would not change its angle with the Earth's perpendicular.

During the race the motorcycle would be able to stop, yet remain balanced without falling over. To keep the vehicle driving down a path at high speeds, however, we employed a technique called 'counter-spin' which works as follows: A riderless motorcycle cannot be turned to the right merely by leaning it to the right; it must first be turned to the left. This maneuver creates a trajectory resulting in a centripetal acceleration which ultimately causes the vehicle to lean to the right. Once leaning at the proper angle, the front wheel of the moltorcycle can be turned to the right to follow a desired trajectory.

Our goal was to combine the two approaches: That is, use the CMG to hold the motorcycle steady while standing still, then decouple the vehicle from the CMG and employ the counter-spin capability for steering once under way.

The MEMS Gyroscope

The second gyroscope we employed is a solid-state, MEMs (micro-electrical-mechanical systems) device. It senses acceleration along the x-y-z axes, and the corresponding roll, pitch and yaw angles — as well as rate of change of each. It is very accurate. However, it does not sense heading.

Inside the enclosure is the GPS card which enables the motorcycle to know its precise position on the map at all times. The control system continuously computes the trajectory the vehicle must follow to stay on course. In fact, we employed a special sensor that uses both GPS and GONAV, a Russian version of GPS. By integrating these systems we obtained greater accuracy than we actually needed.

Absolute accuracy was very important — if we encountered a cliff along the route, say, just 3 feet away, we wanted to make sure our motorcycle was not traveling too close to the edge. That is why a narrow-aperture vehicle was such a good choice. Also, if the route led through an underpass that would be much more easily accomplished with our narrow-aperture motorcycle than with a Hummer.

The motorcycle has electric start, an alternator for battery charging and a 125-cc motor. As far as we know this is the first drive-by-wire motorcycle. And because it is drive-by-wire it is a requisite that the controls be automated. The on-board microcontroller issues the commands that actuate the various controls. There are brake servos that on command, actuate the front and rear brakes. There is also a clutch control that can be engaged or disengaged to change gears, start the vehicle or to bring it to a stop. There is a throttle control.

The vehicle can be started remotely. In the event of a crash and a flooded carburetor, we would be able to flush out the carburetor, actuate the starter via solid-state relays, right the motorcycle and be on our way.

Steering

A controllable 12-VDC motor is used in place of handlebars to steer the vehicle. Linked to it is an optical encoder which continuously monitors the position of the steering shaft. An 80:1 speed reducer multiples the torque developed by the motor by 80 so that the motor develops ample torque to turn the front wheel. Because of this arrangement it is able to turn the front wheel from full left to full right in one-third of a second with a torque of 110 newton-meters.

A set of sensors have been integrated in such a way that the vehicle can successfully perform obstacle detection. The millimeter-range sensor has a range of 110 meters that enables our vehicle to travel at 45 mph and still have six seconds to slow down and come to a stop, if necessary.

We used magnetometers to detect other moving vehicles. These devices detect changes in the Earth's magnetic field that occur when a metallic object enters the vicinity of the motorcycle. The magnetometers give us angle, but not distance. From this we can orient a thermal camera that obtains a thermal image of what is there. If it hot, we then assume that it is an object. If such is the case, we want to avoid getting too close to it.

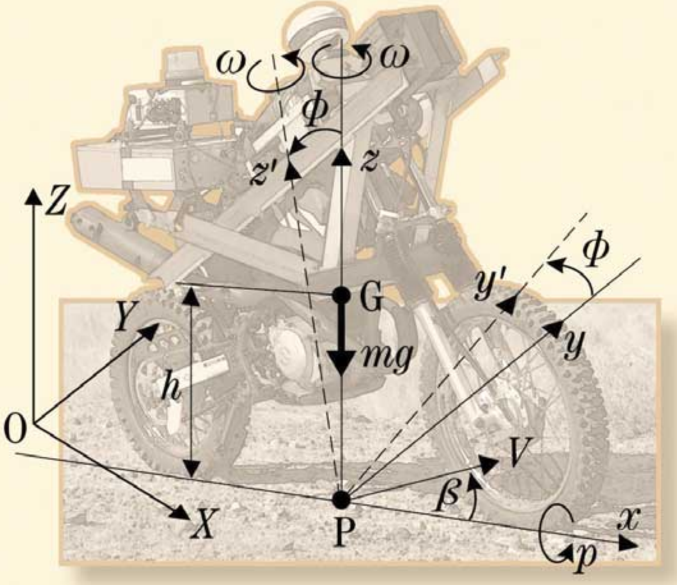
If the vision system thinks it 'sees' a big rock in front of it but the millimeter-wave radar system detects nothing, which do we trust? Do we take a conservative approach and try to go around the obstacle, real or imaginary? So here indeed was the challenge: to decide which of the sensors to trust in the event of conflicting evidence.

As depicted in **Figure 3**, we have modeled the motorcycle as an inverse pendulum. The motorcycle is treated as a pendulum that we balance by controlling the steering of the front wheel. We created a vector that we called X which has the roll angle (ϕ). The steering angle (θ) and the steering angle rate ($\dot{\theta}$). From that we also have the yaw (ω), which is how much the vehicle is turning and the side-slipping angle (β) along the surface.

Using this model we have been able to create a matrix that we can solve, feeding parameters to our program so that it can solve the equation for us and model the control system. The sensors talk via a RS-232 link to our microcontroller and supply the various parameters that are being monitored

Figure 3.

Modeling the motorcycle with a vector as well as the various components that depict, roll, pitch, yaw and slip



ϕ : Roll Angle
θ : Steering Motor Angle
ω : Yawing Angular-Velocity
β : Side Slipping Angle

$$\dot{x} = Ax + Be$$

$$x = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \omega & \beta \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{K_{cf} + K_{cr} - mg}{I_{xx}} h & 0 & \frac{K_f ha}{I_{xx}} & 0 & -\frac{K_f l_f - K_r l_r}{I_{xx}V} h & -\frac{K_f + K_r}{I_{xx}} h \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{b}{J} & 0 & 0 \\ -\frac{K_{cf} l_f - K_{cr} l_r}{I_{zz}} & 0 & \frac{K_f l_f a}{I_{zz}} & 0 & -\frac{K_f l_f^2 + K_r l_r^2}{I_{zz}V} & -\frac{K_f l_f - K_r l_r}{I_{zz}} \\ -\frac{K_{cf} + K_{cr} - mg}{I_{xx}V} h^2 - \frac{K_{cf} + K_{cr}}{mV} & 0 & \frac{K_f h^2 a}{I_{xx}V} + \frac{K_f a}{mV} & 0 & -\frac{K_f l_f - K_r l_r}{I_{xx}V^2} h^2 - \frac{K_f l_f - K_r l_r}{mV^2} - 1 & -\frac{K_f + K_r}{I_{xx}V} h^2 - \frac{K_f + K_r}{mV} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & \frac{k_0}{J} & 0 & 0 \end{bmatrix}^T$$

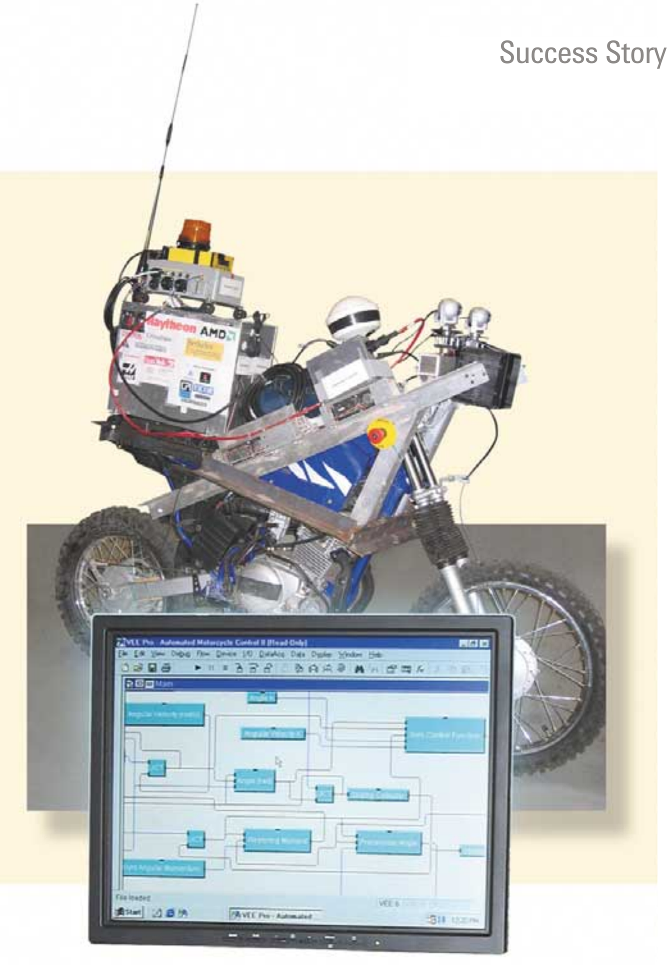


Figure 4.

Modeling the motorcycle in Agilent Technologies' VEE

Choosing a Program

The team used Agilent Technologies' Visual Engineering Environment (VEE) to model the physical system. We had 21 people working on the overall process, each putting in approximately six hours a week. This is probably the most-demanding scenario possible for a development team because nobody has any experience with this specific task.

We chose VEE to model our design because if we had simply begun prototyping directly on a microcontroller, there would have been a tremendous overhead with regard to testing and debugging. The modeling as performed in VEE is depicted in **Figure 4.** Whereas since VEE is all visual, the user receives instant feedback — perhaps an input is missing— or perhaps an obvious error that the user is making in one of his or her assumptions. If we had gone directly to 'C' code we might not have found this out for another month or so.

One way VEE saved us a lot of time is that we didn't have to bring our engineers up to speed to learn C++. With VEE they could dive right and begin interfacing and programming. VEE collects data so we could then appraise just how well our model is working. We have used VEE both on the prototyping side to see how we were going to control the vehicle as well as the development side to decide just which parameters we would need to employ in the model.

It is faster to program in VEE because you can simply type in your formulas and your control algorithms. If you develop it graphically, it virtually solves itself— so there is almost no programming. Virtually anybody can start doing this.

VEE also makes it easy to interface with new sensors. This has been crucial for us because some of the instruments we use are not, say, traditional oscilloscopes or function generators. So we could get down to the very basics and write codes to I/O directly on the sensor layer. Then we could interface with the sensors and send commands directly to them.

VEE actually exceeded our expectations.

If instead of using VEE, the programmers had begun writing our program in 'C', it would have taken two to four weeks just to learn how to do it. But with VEE the entire task required just three hours.

What is Agilent VEE Pro?

Agilent VEE Pro is a powerful, productive, easy to use, graphical programming environment designed to save the user time in building and programming test systems. Agilent VEE Pro is the fastest path from measurement to analysis in existence today.

To create a program, you choose high level graphical objects from the menu and connect them with wires (no more low-level icons to represent each textual line of code). The wire connections help specify functionality and sequences in intuitive block diagrams. You program at a higher, task-oriented level using built-in scientific and engineering routines. Agilent VEE Pro jumpstarts the programming effort and keeps it running smoothly for beginning and veteran programmers alike.

Agilent VEE Pro is designed for productive instrument programming. It can talk to any instrument from any vendor using GPIB, LAN, USB, RS-232, VXI and other common backplanes and protocols. Agilent VEE Pro supports industry standard instrument drivers including IVI-COM and VXI plug&play as well as a variety of legacy drivers. Agilent VEE Pro includes nearly a thousand drivers representing the most popular instruments from 70 different vendors. Agilent VEE Pro can control any standard instrument and many vendor’s PC plug-in cards.

Agilent VEE Pro interfaces with popular office tools: use Microsoft® Word for reports, Excel for spreadsheets, Outlook for paging and e-mail, and Access for database operations. Agilent VEE Pro integrates the .NET Framework and ActiveX, simplifying common tasks and making powerful system capabilities available to your programs. It is easy to complete tasks such as programmatically managing files, sending an email report, or invoking a web page. With the included Active X Automation Server, other programs can call VEE Pro user functions. Agilent VEE Pro accepts data from virtually any vendors’ Instruments, PC cards, and sensors. It supports all popular programming languages including Visual Basic, C/C++, Visual C#, all .NET compliant languages and National Instruments’ LabVIEW. It works with surface-mount machines, robots and other manufacturing equipment via the .NET Framework or ActiveX libraries supplied by the manufacturer. And it can be used as a standalone solution or included in a custom in-house solution. Agilent VEE Pro programs can be monitored and diagnosed remotely via the web or network.

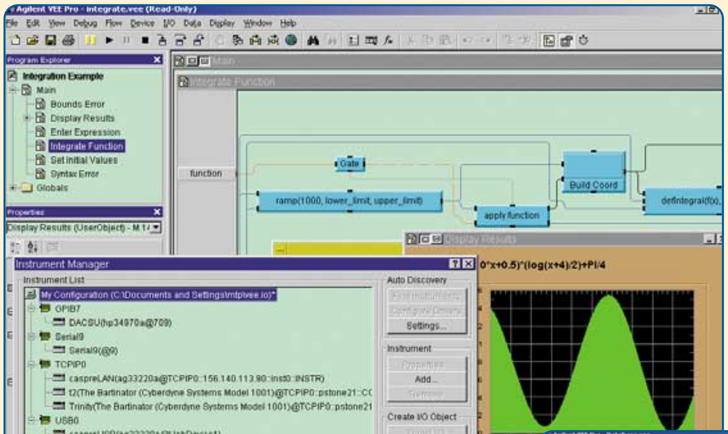
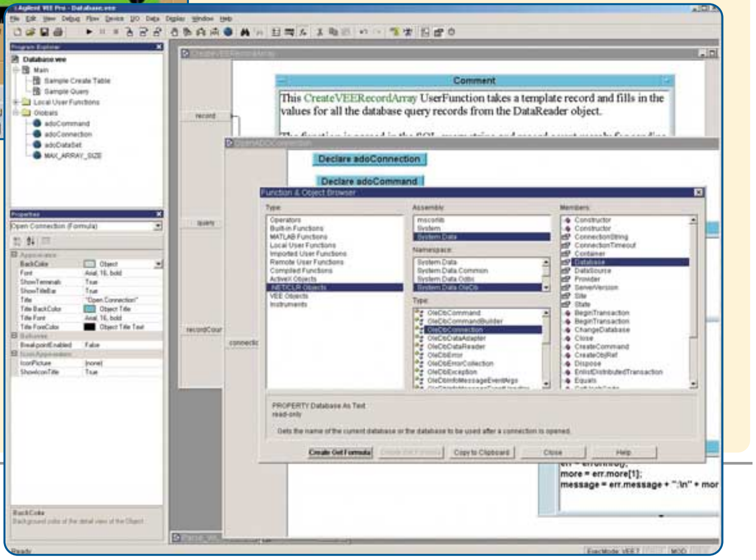


Figure 4.

Agilent VEE Pro optimizes instrument connectivity with its easy to use instrument manager



This CreateVEERecordArray UserFunction takes a template record and fills in the values for all the database query records from the DataReader object.

PROPERTY Database As Text
read-only
Gets the name of the current database or the database to be used after a connection is opened.

```
more = err.more[1];
message = err.message + ":\n" + mor
```

Figure 5.

Agilent VEE Pro utilizes the Microsoft .NET Framework to programmatically access database records

Conclusions

The actual running of the Grand Challenge occured on March 13, 2004. No entry won but there were several that impressed the DARPA agency. Our autonomous motorcycle advanced the state-of-the-art of drive-by-wire, autonomous vehicles and also raised the viability of a two-wheeled vehicle in such a role. What's more, we have proven the viability of VEE as a highly-versatile and powerful program for both modeling and prototyping of sophisticated systems of the sort described above.

Reference

1. *Agilent VEE Pro Multimedia Demonstrations, www.agilent.com*

Anthony Levandowski is the team leader/project investigator. He studied industrial engineering as an undergraduate and electrical engineering in graduate school, specializing in controls. The Grand Challenge was a postgraduate research project for him.

Agilent Email Updates

www.agilent.com/find/emailupdates

Get Free Email Updates

Keep up to date with Agilent's free Email Updates. As a subscriber, you will receive regular, customized email updates on the topics you select. Updates cover support, products and services, applications, promotions, events, and other areas. It is easy to unsubscribe or change your preferences. Subscribe today: **www.agilent.com/find/emailupdates.**

Agilent is committed to respecting and protecting your privacy. Our Privacy Statement at **www.agilent.com/go/privacy** describes our commitment to you. Please direct any questions about Agilent's privacy program to privacy_advocate@agilent.com.

Agilent T&M Software and Connectivity

Agilent's Test and Measurement software and connectivity products, solutions and developer network allows you to take time out of connecting your instruments to your computer with tools based on PC standards, so you can focus on your tasks, not on your connections.
Visit **www.agilent.com/find/connectivity**

Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is available for at least five years beyond the production life of the product. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage."

Our Promise

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, at no extra cost upon request. Many self-help tools are available.

Your Advantage

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and SYSTEMS, and obtain dependable measurement accuracy for the life of those products.

For more information on Agilent Technologies products, applications or services, please contact your local office.

The complete listing is available at:
www.agilent.com/find/contactus

Phone or Fax

United States
(tel) 800 829 4444
(fax) 800 829 4433

Canada
(tel) 1 877 894 4414
(fax) (905) 282 6495

China
(tel) 800 810 0189
(fax) 800 820 2816

Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Korea
(tel) (080) 769 0800
(fax) (080) 769 0900

Latin America
Headquarters:
(tel) (305) 269 7500

Taiwan
(tel) 0800 047 866
(fax) 0800 286 331

Other Asia Pacific Countries:
(tel) (65) 6375 8100
(fax) (65) 6755 0042
(e-mail) tm_ap@agilent.com

Printed in the U.S.A. June 10, 2004
5989-1214EN

Agilent Technologies